ARMY RESEARCH LABORATORY

# A Tool for Visualizing Mobile Ad-Hoc Network Topology Definition Files

by Binh Q. Nguyen

ARL-TR-3750 February 2006

**NOTICES**

**Disclaimers**

# Army Research Laboratory
Adelphi, MD 20783-1197



**ARL-TR-3750** **February 2006**


# A Tool for Visualizing Mobile Ad-Hoc Network Topology Definition Files


**Binh Q. Nguyen**
**Computational and Information Sciences Directorate, ARL**

| **REPORT DOCUMENTATION PAGE** | *Form Approved* *OMB No. 0704-0188* |
|---|---|



| | | |
|---|---|---|
| **1. REPORT DATE** *(DD-MM-YYYY)*<br>February 2006 | **2. REPORT TYPE**<br>Final | **3. DATES COVERED (From - To)**<br>01 Oct 04 – 31 Sep 05 |
| **4. TITLE AND SUBTITLE**<br>A Tool for Visualizing Mobile Ad-Hoc Network Topology Definition Files | | **5a. CONTRACT NUMBER** |
| | | **5b. GRANT NUMBER** |
| | | **5c. PROGRAM ELEMENT NUMBER** |
| **6. AUTHOR(S)**<br>Binh Q. Nguyen | | **5d. PROJECT NUMBER**<br>5FE6BG |
| | | **5e. TASK NUMBER** |
| | | **5f. WORK UNIT NUMBER** |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>U.S. Army Research Laboratory<br>ATTN: AMSRD-ARL-CI-CN<br>2800 Powder Mill Road<br>Adelphi, MD 20783-1197 | | **8. PERFORMING ORGANIZATION REPORT NUMBER**<br>ARL-TR-3750 |
| **9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**<br>U.S. Army Research Laboratory<br>2800 Powder Mill Road<br>Adelphi, MD 20784-1197 | | **10. SPONSOR/MONITOR'S ACRONYM(S)** |
| | | **11. SPONSOR/MONITOR'S REPORT NUMBER(S)** |

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release, distribution unlimited

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

This report documents a research and development effort that the U.S. Army Research Laboratory conducted in FY05 in support of the building of an emulated tactical mobile ad hoc network (MANET) research test bed. The effort resulted in the creation of two visually appealing tools called *TDFvisualizer* and *TDFanimator*. The main purpose of the tools is to provide researchers with a method for visually inspecting and verifying sequences of changing MANET topologies specified in a topology definition file (TDF), a method for conducting dry runs before emulating a MANET in a physical network, a method for graphically illustrating the behavioral complexity of a MANET to unfamiliar users for their education and training, and a method for graphically demonstrating to actual and potential customers and supporters the capability of the test bed and the emulation of a MANET.

**15. SUBJECT TERMS**

Mobile ad hoc network, topology emulation, visualization, animation

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON**<br>Binh Q. Nguyen |
|---|---|---|---|---|---|
| **a. REPORT**<br>Unclassified | **b. ABSTRACT**<br>Unclassified | **c. THIS PAGE**<br>Unclassified | UL | 34 | **19b. TELEPHONE NUMBER** (*Include area code*)<br>301-394-1781 |

**Standard Form 298 (Rev. 8/98)**
**Prescribed by ANSI Std. Z39.18**

## Contents

## List of Figures

## Preface

This work was performed in support of the building of an emulated mobile ad hoc network research test bed at the U.S. Army Research Laboratory, Adelphi, Maryland. The test bed is a networked environment in which routing protocols and applications are developed, tested, evaluated, and demonstrated.

# Executive Summary


The U.S. Army Research Laboratory (ARL) has been conducting research in mobile ad hoc networks (MANETs) with external researchers though the Collaborative Technology Alliances (CTA) program for several years.  As ARL is building an emulated MANET research test bed, the visualization and animation tools were developed to support this effort.  The tools are used for visually inspecting, verifying, demonstrating, and reviewing a desired topology.

An emulation of a MANET at ARL is accomplished by using a software tool called *tsm* to create logical topologies for an ordinary local area network.  Executing the *tsm* tool requires the availability of an input topology definition file (or simply a *tdf* file).  Each *tdf* file contains sequences of topology definitions expressed in the scenario definition language (SDL).  The *tsm* tool and SDL were developed by Telcordia Technologies, Inc., a CTA research member.

To display accurately the positions of each MANET node on a computer screen requires position location information, but the original SDL does not include any rules to specify a location for each node.  Therefore, ARL has extended the SDL grammars to include a way to express position location information and created graphical software tools that can display the position of each MANET node on the screen as intended by topology designers.  The tools include the *TDFvisualizer* tool and the *TDFanimator* tool.  The latter is the successor of the former.  The former is designed to visualize a fixed-sized MANET.  The latter can deal with changing-sizes of a MANET, and it provides the user with a method for controlling the animation manually.

The tools have played an essential role in the design and the verification of desired topologies required for the emulation of an ad hoc network at ARL and at Telcordia Technologies, Inc.  The successful creation of the tools has enabled the design, visualization and animation of complex topologies for an emulated MANET.  The employment of ARL-developed visualization tools at CTA laboratories attests to the usefulness of the tools and to the contribution of ARL. The tools provide MANET researchers with at least the following methods for visualizing and animating scenario definition files:

(a) A method for verifying the accuracy of a topology scenario by visually inspecting a *tdf* file instead of reading through sequences of textual topology descriptions.

(b) A method for visually conducting dry-runs of an emulated topology.

(c) A method for inspecting large *tdf* files through visualization and animation. Without the visualization tools, manually tracking the changes of a dynamic topology would be an extremely time-consuming and difficult task.

(d) A method for graphically illustrating the behavioral complexity of a MANET to unfamiliar users for their education and training.

(e) A method for graphically demonstrating to actual and potential customers and supporters the capability of the test bed and the emulation of a MANET.

The real impact of the visualization and animation tools on its users is the notable increases of their productivity and satisfaction. Using these tools, CTA researchers were, for the first time, able to systematically visualize and verify topology definition files quickly and reliably; therefore, they could save time and effort, reduce complexity, and avert errors.

# 1. Introduction

## 1.1 Background

The U.S. Army Research Laboratory (ARL) has been conducting research in mobile ad hoc networks (MANETs) with industry and academia partners though the Collaborative Technology Alliances (CTA) program for several years. Telcordia Technologies, Inc. (hereafter Telcordia) and MacAfee Research (now part of SPARTA, Inc.) have been MANET research partners of ARL. A MANET is an autonomous system consisting of cooperative devices called nodes capable of organizing themselves into a network without relying on any centralized services (*1*). Mainly due to mobility, the topology of a MANET can eventually change. Dynamic topology is a distinguishing feature of a MANET.

In the summer of 2003, ARL initiated an effort to build a research test bed environment in which MANET technologies developed by the CTA partners could be effectively developed, tested, evaluated, and demonstrated. As Telcordia had already built a test bed in its tactical environment assurance laboratory (TEAlab) (*2*), ARL adopted the proven tools and methods used in the TEAlab for the building of its own internal research test bed. During a visit to the TEAlab, ARL representatives were shown the tools and methods for building an emulated MANET using an ordinary network of computers running the Linux operating system. The learning from this visit includes the following observations:

- An emulated topology of a MANET can be accomplished by executing a packet filtering software called *iptables* (http://www.netfilter.org) that can drop or accept an incoming data packet, effectively creating a logical topology for a network
- The emulated topology is controlled by the topology scenario management *(tsm)* tool that takes in a topology definition file (*tdf*), also known as scenario definition file (*sdf*), interprets and translates sequences of topology directives specified in *tdf* files into *iptables* commands
- The topology directives are specified in the scenario definition language (SDL). Each networked computer in the TEAlab runs the Ad-Hoc on Demand Distance Vector (AODV) routing protocols
- The links among the participated MANET nodes are monitored and displayed in real time using a proprietary topology visualization tool called “AODV Monitor,” which automatically places the nodes on the screen and the connections among them

- An off-line graphical tool, called the communication scenario generator (CSG) tool, is capable of generating topology directives and storing them in a text file called "visibility database (VDB)" (*3*)
- The CSG-generated topology directives are based on the calculated line-of-sight (LOS) visibility between any two nodes
- A *tdf* file can be manually created using a text editor or derived from CSG-created VDB files.

Telcordia subsequently licensed and delivered to ARL the *tsm* tool, version 1.2, in September, 2003, and the CSG tool, version 3.5.8, in August, 2004. The real-time AODV Monitor visualization tool was not licensed to ARL.

## 1.2 Topology Definition Files

The emulation of a MANET using tools and techniques developed by Telcordia requires the availability of *tdf* files. An important lesson learned from dealing with *tdf* files is that the rules for creating them can be expressed in the following Backus-Naur form (BNF):

| | |
|---|---|
| < *tdf*-file> | ::= <topology-definition>+ |
| <topology-definition> | ::= <topology-directive>+ <time-duration> |
| <topology-directive> | ::= "on" <receiver> <action> <sender> "inbound" "\n" |
| <receiver> | ::= computer-name \| host-name |
| <action> | ::= "deny" \| "accept" |
| <sender> | ::= computer-name \| host-name |
| <time-duration> | ::= "wait for" <an-integer> "seconds" "\n" |

The rules state that each *tdf* file contains the definition of at least one topology. Each topology has at least one topology directive and a time-duration statement. A dynamic MANET is emulated by combining the specifications of multiple emulated topologies into a single *tdf* file and calling it a dynamic scenario file (hereafter used as a DSF file).

The number of directives in a *tdf* file is a quadratic function, $f(m,n)$, that depends on the size $(n)$ of the MANET under test and the number of emulated topologies $(m)$. The number of participating computers in an emulated MANET determines the size of the MANET under test. As an emulated topology for an $n$-node MANET has $n$-1 topology directives for each node, the total number of lines in a *tdf* file, $p(n)$, of an $n$-node MANET, is $n(n-1)$. If a *tdf* file contains $m$ distinct topologies to emulate a dynamic scenario, the total number of lines of SDL code requires $m \cdot n(n-1)$ topological directives; i.e., $f(m,n) = m \cdot n(n-1)$.

As mathematically demonstrated, the size and the complexity of SDL code exponentially increase as the size of an emulated MANET linearly increases. Therefore, a *tdf* file is often shortened by including only the specifications of topological changes between two consecutive topologies.

In summary, topology definitions are expressed in a high-level language called SDL and stored in human-readable text files. The SDL language enables a topology designer to specify an action to be performed on a link between any two nodes, and the time duration of each topology.

### 1.3 Visualization Problems and Related Work

During FY04, ARL and its collaborative research partners unanimously expressed a desire to be able to view graphical representations of MANET topologies defined in *tdf* files. The visualization of textual descriptions would benefit them in many different ways. For example, the visualization would assist them in better comprehending emulated network topologies and provide them with a method for visually inspecting, verifying, and demonstrating a dynamic scenario by visualizing and animating *tdf* files.

As ARL was building an emulated MANET test bed environment in which collaborative research activities could be conducted, ARL was also responsible for providing the researchers with a needed visualization tool. The initial approach was to use the CSG tool for visualization purposes, but after an empirical experiment with the tool, ARL concluded that the tool could not be used for the visualization of *tdf* files because of many technical problems. Two foremost visualization-related problems were that the CSG-created VDB files contained no position information and that the tool could not load an arbitrary *tdf* file for visualization purposes.

Subsequently, ARL conducted an assessment of existing visualization tools designed specifically for MANETs to find an appropriate tool potentially capable of visualizing any *tdf* files. ARL found several visualization tools, but it could not use any of the available tools because not one of them could take in a *tdf* file. Nevertheless, the assessment provided ARL with the information about the functionality and capability of MANET visualization tools should they be needed in the future.

The assessed visualization tools can be classified in two categories: position plotting tools and graph visualization. The former requires position information, and the latter does not. The position plotting tools for MANETs consist of three tools: *Jmap* (*4*, *5*), Mobility Visualizer (*6*), and *MobiEmu* (*7*) tools. The *Jmap* tool displays in real time the topology of a MANET under test using global positioning system (GPS) and internal routing information. The Mobility Visualizer displays mobility scenarios for ad-hoc networks generated by the *setdest* tool (*8*). The *MobiEmu* tool displays mobility scenarios for ad-hoc networks generated also by the *setdest* tool.

The assessed graph-visualization tools for MANET include three tools: *mNet* (*9*), Dot Topology Information plugin (*10*), and AODV Monitor (*2*). The *mNet* tool generates and displays the topology of a MANET by applying an automatic layout algorithm to display arbitrary mobility scenarios. The Dot Topology Information plug-in is designed to work with the Optimized Link State Routing (OLSR) protocol software to display the dynamic topology of a MANET in execution. The AODV Monitor is a proprietary real-time graph visualization tool used in the TEALab of Telcordia to display dynamic topology of a MANET running the AODV routing protocols.

The automatic layout algorithm provides a convenient way for displaying the links among the connected MANET nodes, but it does not place the graphical representations of MANET nodes on their intended locations on the screen. These tools are adequate for visually inspecting the links among the nodes, but inadequate for depicting the geographical shape of the topology being visualized.

Therefore, the lack of a suitable tool and the lack of position information in *tdf* files had precluded ARL and its research partners from graphically inspecting their designed topologies until ARL successfully created the *TDFvisualizer* tool and its successor, the *TDFanimator* tool.

### 1.4 Purpose and Scope of the Report

This report documents two unprecedented visualization tools called *TDFvisualizer* and *TDFanimator* that have been independently designed and developed by ARL. The tools have been used by internal and external researchers to visualize *tdf* files designed for the emulation of a MANET since November, 2004. This report has three main purposes:

- Reporting a successful development of two visualization tools that the Battlefield Communication Networks Branch has contributed to the mission of ARL
- Describing the features, benefits, and capabilities of the visualization tools
- Presenting plans for further improving the visualization tools

The intended readers of this report include practicing MANET researchers, engineers, and technical managers.

The next section briefly presents the emulation of a MANET at ARL and how the visualization tools are used in the process. Section 3 explains the design philosophy of the tools. Section 4 describes the inputs and the outputs of the tools, including its features, functionality, and capabilities. Section 5 presents and discusses the results of the use of the tools and a plan for improving them. Section 6 concludes the report by summarizing and highlighting the benefits of the tools.

## 2. Emulation Process

Although the MANET emulation lab at ARL is very similar to the TEAlab at Telcordia, ARL has improved its capabilities by extending the topology file formats, independently developing automated tools, and using real-time communication and visualization tools developed by the Naval Research Laboratory (NRL). The extended file formats include the position information *(pos)* and the composite topological information *(rgp)* formats. The position information is expressed in pixel coordinates relative to a visualization tool. The composite topological information combines the topology directives and the position information into the same file. ARL has also modified the *tsm* tool to enable it to deal with position information, which is required for accurately placing graphical presentations of MANET nodes on a visualization tool. Appendix A provides a sample of these data files.

The emulation process has two phases. The first phase is conducted off-line using an internally developed tool set to visually create *tdf* files required for the emulation of a MANET. The *tdf* visualization tools, *TDFvisualizer* and *TDFanimator*, are part of the tool set. The second phase actually emulates a dynamic MANET in a physical network using the ARL-modified *tsm* tool and the NRL-developed *Nettion* tool set. The *tsm* tool is used to execute a *tdf* file to create logical topologies for the emulation of a MANET. The *Nettion* tool set, including the *Jmap* visualization tool, is used for real-time data analysis and visualization purposes. The two phases of the emulation process are depicted in figure 1. The visualization process is encircled to emphasize the role of the visualization and animation tools in the creation of *tdf* files during the first phase.

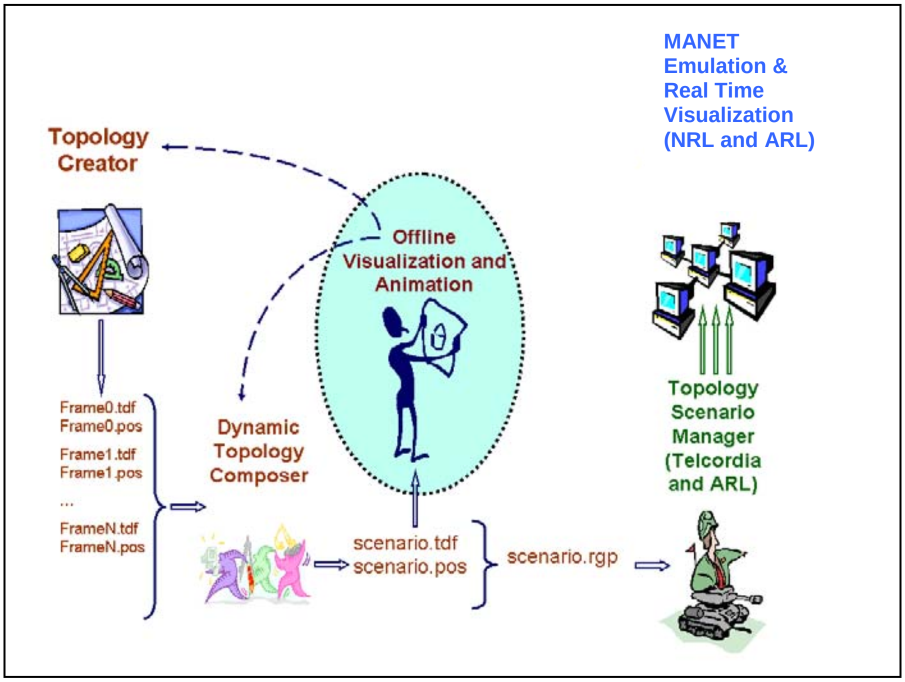



Figure 1. The role of visualization and animation tools in the emulation of a MANET.

As the focus of this report is on the visualization and animation tools, detailed descriptions of the actual emulation of a MANET and other tools used in the process are beyond the scope of this document.

## 3. Design Principles

The visualization tools were created to meet the pressing needs of CTA researchers with a set of principles that form the basis and the requirements for their existence. The principles include simplicity, usability, portability, reusability, and adaptability. To realize these principles, ARL adopted the object-oriented Python programming language and its standard graphical user interface toolkit called *Tkinter* (*11*) to build the visualization and animation tools.

### 3.1 Simplicity

The visualization tools perform a single function: graphically rendering the textual topology descriptions stored in *tdf* files. They perform no other functions, except providing their users with a graphical interface for selecting a *tdf* file to visualize.

### 3.2 Usability

As the visualization tools were created in response to a pressing need, they were designed and developed with the focus on the need of its users. The tools have been frequently used by researchers for many purposes. The foremost purpose is to visually verify their designed topologies, effectively conducting dry-runs before actually emulating a MANET. Other purposes include the demonstration of a dynamic MANET to visitors, management, and actual and potential customers.

### 3.3 Portability

Although the visualization tools were developed and tested in the Microsoft Windows XP environment, they run in the Linux operating systems without any special adjustments. The ability of the tools to run in dissimilar operating environments minimizes the time required to develop, install, configure, and run the tools. This beneficial feature of the tools can be attributable to the portability of the Python code.

### 3.4 Reusability

As Python is an object-oriented computer language, previously implemented code can be reused or inherited to minimize the effort and the time required to add a new functionality. For example, the *TDFanimator* tool was implemented as an extension of the *TDFvisualizer* tool.

### 3.5 Compatibility

The visualization tools can take in various file formats; i.e., *tdf, pos,* and *rgp*. The *tdf* file format is created to meet the requirements of the original *tsm* tool. The *pos* format is used for placing a graphical presentation of a MANET node on the canvas of the *TDFvisualizer* tool, the predecessor of the *TDFanimator* tool. The *rgp* format is created to satisfy the requirements of the ARL-enhanced *tsm* tool. This format combines the other two formats (*tdf* and *pos*) into a single file.

### 3.6 Adaptability

The visualization tools are adaptable to changing requirements as the existence of the *TDFanimator* tool attests to this ability. Because the tools are internally developed, they can be improved at any time to meet changing requirements without first obtaining a license from any third party. As the nature of research and development often changes, the tool is anticipated to be proportionally changed accordingly.

# 4. Visualization Tools

The *tdf* visualization tools relate to networked communications, especially stationary and mobile ad hoc networks, and principally to a method for visually inspecting and verifying topology definitions expressed in SDL. The time required to learn how to use the tools is very short as the tools incorporate graphical user interfaces and a familiar computer pointing device.

The two visualization tools are: *TDFvisualizer* and *TDFanimator*. Both tools are implemented using the Python programming language. They were designed and developed at different times to meet different needs, and they differ from one another in terms of visualization features and the internal processing of *tdf* files. The history and the details of these two tools will be described in the subsequent sections.

## 4.1 History

Conceived sometime in August, 2004, and documented in a FY05 task plan in September, 2004, the first prototype of the *TDFvisualizer* tool was successfully developed and demonstrated to ARL and MacAfee Research engineers and managers in November, 2004 at a McAfee facility, Rockville, MD. The tools capability was well received by the audience, so the development of the tool continued until January, 2005, when the tool was released to ARL engineers for operational use.

Enabling the tool to load iconographic images was the final incremental improvement to the *TDFvisualizer* tool as ARL was preparing for a demonstration to visitors in May, 2005. The use of images did not add any new functional capability to the tool, instead it added visual appeal and ease of understanding. The tool and the source code were released to Telcordia researchers by an ARL engineer in May, 2005.

At the end of May, 2005, although the *TDFvisualizer* tool was running as intended, its source code was reused (inherited) to create a new tool called *TDFanimator* to satisfy the following new requirements requested by its users:

- Playing forward or backward frame by frame, manually or automatically
- Abruptly stopping, then proceeding with the visualization in progress
- Displaying the first frame or the last frame (rewinding or forwarding)

The *TDFanimator* tool was completed and sent Telcordia on June 20, 2005. Users of the visualization tools now have the option to use either the *TDFvisualizer* tool or the *TDFanimator* tool, depending on individual preferences.

### 4.2 The TDFvisualizer Tool

The *TDFvisualizer* tool was created to provide MANET researchers participating in the CTA program with a method for visualizing a *tdf* file. The tool was conceptualized and implemented because no commercially available or open-source tools could load *tdf* files and visualize their textual descriptions expressed in SDL. As the early tdf files were manually created to emulate the dynamic topology of a small, fixed-sized MANET, consisting of 6 to 10 nodes, the *TDFvisualizer* tool was designed to handle the constraints imposed by these *tdf* files.

As *tdf* files hold only topology directives expressed in SDL, no visualization tools could render a topology on the screen as intended due to lack of position location information. Therefore, the desire to place a MANET node at its intended position on the screen necessitated the creation of position location information for each MANET node.

The newly created position information is stored in a separate file having a *'.pos'* file extension. Each position statement is expressed in two dimensional pixel coordinates and in the format of a comma-separated, three-element tuple: *<node-name>, <x>, <y>*. For example, the statement *"alpha, 230, 350"* indicates that the horizontal and vertical positions of the MANET node named *alpha* are *230* pixels from the left and *350* pixels from the top of the display area of the *TDFvisualizer* tool

The separating of the position information ensured the operability of the original *tsm* tool by keeping the original *tdf* files intact. (The *tsm* tool could not deal with any position location information.) Therefore, each *tdf* file has an associated *pos* file. The two files have the same base name, taken from the name of the *tdf* file; *e.g., xyz.tdf* and *xyz.pos*.

In January, 2005, an ARL engineer modified the *tsm* tool to handle the position location information and, therefore, created a new position information format that can be included in a topology scenario file, effectively forming a new type of data file called "*rgp*" (which is the three initials of its creator). The new position location information format is: *"on <node> position <x> <y>"*. This modification eliminates the need for having a *pos* file for each *tdf* file by creating an *rgp* file that combines the current *tdf* data format with the new position format. This change necessitated the enhancement of the *TDFvisualizer* tool to handle the new data format of *rgp* files.

The tool can load three different types of input data files. Given the name of an input *tdf* file, the tool automatically extracts the base name from the full name, searches for a *pos* file having the same base name, loads the associated *pos* file, then starts rendering the graphics on the screen until the scenario ends. Similarly, if the given name is a *pos* file, then the tool automatically searches for a corresponding *tdf* file. Given the name of an *rgp* file, the tool immediately loads the file without searching for any other files. Appendix A provides a sample of these data files.

To obtain an input file, the tool uses a graphical dialog to ask the user for the name of an input file.  The user then can select a file from the displayed list of files or type in a *pos*, *tdf*, or *rgp* file as depicted in figure 2.

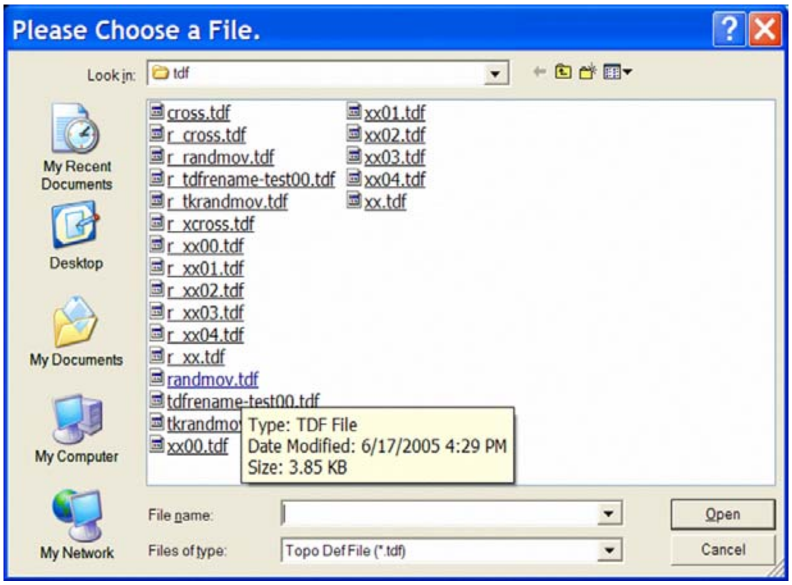



Figure 2.  Graphical dialog for selecting an input topology definition file.

Once the visualization has started, it cannot be gracefully stopped.  The only way to abruptly stop the visualization in progress is by clicking the red symbol "[X]" located at the upper right of the window or by clicking the *Close* button located at the lower right corner of the window.  Figure 3 displays two MANET topologies being rendered in the *TDFvisualizer* tool.  Each MANET node in the figure is represented by an iconographic image of a tank, and if a link between any two nodes exists, then it is represented by an arrow connecting the two nodes.  The head of the arrow points to the receiver, and the tail rests on the transmitter.


Figure 3. The *TDFvisualizer* tool.

As seen in figure 3, two ARL icons are displayed on the upper left and the upper right corners of the screen. These icons serve a single purpose: indicating the ownership of the visualization tool. The bottom of the screen is a row of six buttons, being labeled as *XY-Print*, *TDF-Print*, *TDF-Create*, *TDF-Load*, *Conf-Print,* and *Close*. The functions of these buttons are:

- XY-Print — print the position location information of each node to the console. This feature is created to assist in the development of the tool and to ensure its integrity.

- *TDF-Print* — print the link information (topology directives) of each node to the console. This feature exists to assist in the development of the tool and to ensure its consistency.

- *TDF-Create* — automatically create a random scenario for testing purposes. This feature exists to assist in the development of the tool and to ensure its integrity. It is now disable due to the abundance of data files.

- *TDF-Load* — open a dialog window to ask the user for the name of an input file. The user then can select a file from the displayed list of files or type in a *pos*, *tdf*, or *rgp* file as depicted in figure 2.

- *Conf-Print* print the configuration information of the tool to the console. This feature exists to assist in the development of the tool and to ensure its reliability.
- *Close* exit the tool.

The *TDFvisualizer* tool can be configured to run with or without images, which include a background image and an iconographic image representing a MANET node. Some of the aesthetic symbols and attributes of the tool are user-configurable and stored in a configuration file. By editing the configuration file, the user can specify certain options and preferences, including the following items:

- The width and the height of the canvas in terms of number of pixels
- The name of a background image file or the background color of the screen
- The name of an image file representing a node (e.g., vehicle, soldier) or the color and size of a graphical representation of a node (e.g., a circle)
- The color of a link connecting any two nodes

In summary, the *TDFvisualizer* tool has the following features and requirements:

- The size of a MANET must remain constant throughout the emulation
- Each visualization requires an *rgp* file or a *tdf* file and a *pos* file
- Topology definition is expressed in SDL
- Position location information are expressed in pixel coordinates having the origin (0, 0) located at the upper left corner of the screen.

### 4.3 The TDFanimator Tool

The *TDFanimator* tool was created to satisfy the requirements requested by the users of the *TDFvisualizer* tool. It inherits all the functionality of the *TDFvisualizer* tool and adds the following features and capabilities:

- Playing forward or backward frame by frame, manually or automatically
- Being able to be abruptly stopped then proceeding with the visualization in progress
- Displaying the first frame or the last frame (rewinding or forwarding)

Figure 4 depicts two screen images of the *TDFanimator* tool while it was visualizing and animating a scenario file specifying the dynamic movement of an emulated MANET.


Figure 4. The *TDFanimator* tool.

The bottom of the tool contains a set buttons being labeled as *Open, Open+Run,* |<, <<, *Play*|*Stop,* >>, >|, and *Close*. The functions of these buttons are described below:

- *Open* open a dialog window to ask the user for the name of an input file. The user then can select a file from the displayed list of files or type in a file name as depicted in figure 2.
- *Open+Run* load a *pos*, *tdf*, or *rgp* file and immediately start the animation.
- |< display the first topology.
- << animate the scenario in the reverse direction.
- *Play* | *Stop* play a loaded scenario file or stop an on-going animation.
- >> animate the scenario in the forward direction.
- >| display the last topology.
- *Close* exit the tool.

The *TDFanimator* tool thus operates like a video player providing the user with various options for visualizing and animating a topology definition file.

## 5. Results and Discussion

The employment of ARL-developed visualization tools at CTA laboratories attests to the usefulness of the tools and to the contribution of ARL. The tools provide MANET researchers with at least the following methods for visualizing and animating scenario definition files:

- A method for verifying the accuracy of a topology scenario through visual inspection and animation of topology definition files instead of manually examining sequences of textual descriptions of topology definition.
- A method for visually conducting dry-runs by previewing sequences of changing topologies through visualization and animation of topology definition files.
- A method for inspecting large topology definition files specifying a dynamic topology of a large MANET through visualization and animation. Without the visualization tools, manually tracking the changes of a dynamic topology would be an extremely time-consuming and difficult task.
- A method for graphically illustrating the behavioral complexity of a MANET to unfamiliar users for the purpose of education and training.
- A method for graphically demonstrating to actual and potential customers and supporters the capability of the test bed and the emulation of a MANET.

Future improvements to the tools will include at least the following features:

- latitude-longitude coordinates
- color-coded link quality
- popup menus

# 6. Conclusions

The successful creation of the *TDFvisualizer* tool and the *TDFanimator* tool has enable the visualization and animation of complex topology of an emulated MANET. Conducting dry-runs to visually verify the dynamics of an emulated MANET provides researchers with an effective way for inspecting a topology definition file before emulating a MANET using a physical network test bed. The tools are playing an essential role in the design and the verification of desired topologies required for the emulation of an ad hoc network.

The real impact of the visualization and animation tools on its users is the notable increase in their productivity and satisfaction. The tools are being used not only within ARL, but also at Telcordia Technologies—a premier research institution and a CTA member of ARL. Using these tools, CTA researchers were, for the first time, able to systematically visualize and verify topology definition files quickly and reliably; therefore, they could save time and effort, reduce complexity, and avert errors.

# 7. References


1. Corson, S.; Macker, J. Mobile Ad hoc Networking (MANET): Routing Protocol Performance: Issues and Evaluation Considerations, *The Internet Engineering Task Force (IETF) Request for Comment (RFC2501)*, January 1999. URL: http://www.ietf.org/rfc/rfc2501.txt?number=2501 (access date: 08 Sep 05).
2. Little, Mike. TEALab: A Testbed for Ad-hoc Networking Security Research, *Proceedings of MILCOM2005*, Atlantic City, NJ, October 17-20, 2005.
3. Science Applications International Corporation (SAIC) Research and Development Division – Co. 1, "COMTEST Scenario Generator User's Manual and Technical Reference for the Version 3.4," SAIC, 10260 Campus Point Drive, San Diego, California 92121, c/o 1410 Springhill Road, Suite 400, McLean, VA 22102, 1 October 2003.
4. Makcer, J. P.; Chao, W.; Weston, J. W. A Low-Cost, IP-Based Mobile Network Emulator (MNE). *Proc. Military Communications Conference, IEEE MILCOM 2003*, October 2003, pp. 481-486.
5. The U.S. Naval Research Laboratory, *Nettion* (NETwork TestIng and Operational eNvironment). URL: http://nettion.pf.itd.nrl.navy.mil/nettion_index.html (accessed 10 September 2005).
6. Workgroup Mobile Ad-hoc Networks, Mobility Visualizer, Lehrstuhl für Informatik 4 Aachen University. URL: http://www-i4.informatik.rwth-aachen.de/~mesut/manet/tools/MobilityVisualizer.jar.
7. Zhang, Yongguang; Li, Wei. An Integrated Environment for Testing Mobile Ad-Hoc Networks. *Proc. Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02),* Lausanne, Switzerland, Jun 2002.
8. The Carnegie Mellon University Monarch Project's Wireless and Mobility Extensions to ns, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, August 5, 1999.
9. Karygiannis, A.; Antonakakis, E. mLab: a Mobile Ad Hoc Network Test Bed, 1st Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing in conjunction with the IEEE International Conference in Pervasive Services 2005, July 14, 2005, Santorini, Greece. URL: http://csrc.nist.gov/manet/SecPerU2005-Karygiannis-Antonakakis.pdf (accessed 10 September 2005).

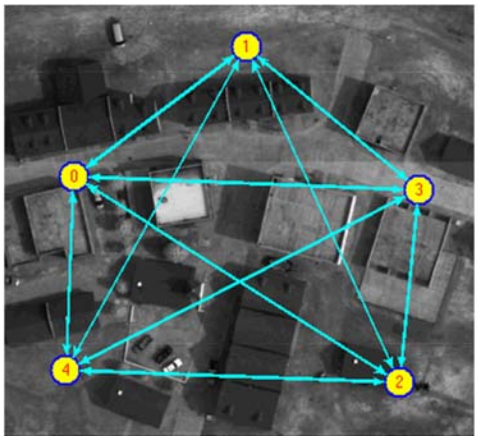
10. Tønnesen, Andreas. “Dot Topology Information Plugin,” URL: http://www.olsr.org.

11. The Python Software Foundation, “What is Python?” URL: http:www.python.org (accessed 11 September 2005).

INTENTIONALLY LEFT BLANK.

## Appendix A. Topology Scenario Files

The contents of topology scenario files specifying the star topology are shown below:




***tdf* file**

```
on 0 accept 1 inbound
on 0 accept 4 inbound
on 0 accept 2 inbound
on 0 accept 3 inbound
on 1 accept 0 inbound
on 1 accept 3 inbound
on 1 accept 2 inbound
on 1 accept 4 inbound
on 2 accept 4 inbound
on 2 accept 3 inbound
on 2 accept 1 inbound
on 2 accept 0 inbound
on 3 accept 2 inbound
on 3 accept 1 inbound
on 3 accept 0 inbound
on 3 accept 4 inbound
on 4 accept 0 inbound
on 4 accept 2 inbound
on 4 accept 3 inbound
on 4 accept 1 inbound
wait for 1 seconds
```

***pos* file**

```
0, 165, 242
1, 310, 132
2, 435, 420
3, 462, 237
4, 181, 405
wait for 1 seconds
```

***rgp* file**

```
on 0 position 165 242
on 0 accept 1 inbound
on 0 accept 4 inbound
on 0 accept 2 inbound
on 0 accept 3 inbound
on 1 accept 0 inbound
on 1 accept 3 inbound
on 1 accept 2 inbound
on 1 accept 4 inbound
on 2 accept 4 inbound
on 2 accept 3 inbound
on 2 accept 1 inbound
on 2 accept 0 inbound
on 3 accept 2 inbound
on 3 accept 1 inbound
on 3 accept 0 inbound
on 3 accept 4 inbound
on 4 accept 0 inbound
on 4 accept 2 inbound
on 4 accept 3 inbound
on 4 accept 1 inbound
on 1 position 310 132
on 2 position 435 420
on 3 position 462 237
on 4 position 181 405
wait for 1 seconds
```

INTENTIONALLY LEFT BLANK.

## Appendix B.  Acronyms and Abbreviations

| | |
|---|---|
| COMTEST™ | A suite of software tools designed for testing communications systems developed by Science Applications International Corporation (SAIC) |
| Iptables | An open-source packet filtering and mangling software available in the Linux operating systems.  URL:  http://www.netfilter.org/ |
| MANET | Mobile ad hoc network – an autonomous system of collaborative computing platforms that can function without relying on an establish network infrastructure.  URL: http://www.ietf.org/rfc/rfc2501.txt |
| POS | Position information in pixel coordinates relative to the canvas area of the tool |
| SDL | A scenario definition language developed by SAIC-Telcordia Technologies.  It is used to define an emulated topology of a MANET |
| TDF | Topology definition file format defined by Telcordia Technologies.  It is a text file containing sequences of topological directives expressed in the scenario definition language (SDL) |
| *TDFanimator* | The visualization and animation tool is the successor of the *TDFvisualizer* tool |
| *TDFvisualizer* | The visualization and animation tool that provides a way to visualize and animate topology definition files. |
| RGP | The topology data format defined by Richard Gopaul of Army Research Laboratory, Adelphi, Maryland |
| LOS | Line of sight |
| VDB | Visibility database (generated by the CSG tool) |

INTENTIONALLY LEFT BLANK.

## Distribution

ADMNSTR
DEFNS TECHL INFO CTR
ATTN DTIC-OCP (ELECTRONIC COPY)
8725 JOHN J KINGMAN RD STE 0944
FT BELVOIR VA 22060-6218

DARPA
ATTN IXO S WELBY
3701 N FAIRFAX DR
ARLINGTON VA 22203-1714

OFC OF THE SECY OF DEFNS
ATTN ODDRE (R&AT)
THE PENTAGON
WASHINGTON DC 20301-3080

US ARMY TRADOC
BATTLE LAB INTEGRATION & TECHL
DIRCTRT
ATTN ATCD-B
10 WHISTLER LANE
FT MONROE VA 23651-5850

SMC/GPA
2420 VELA WAY STE 1866
EL SEGUNDO CA 90245-4659

US ARMY ARDEC
ATTN AMSTA-AR-TD
BLDG 1
PICATINNY ARSENAL NJ 07806-5000

COMMANDING GENERAL
US ARMY AVN & MIS CMND
ATTN AMSAM-RD W C MCCORKLE
REDSTONE ARSENAL AL 35898-5000

US ARMY INFO SYS ENGRG CMND
ATTN AMSEL-IE-TD F JENIA
FT HUACHUCA AZ 85613-5300

US ARMY SIMULATION TRAIN &
INSTRMNTN CMND
ATTN AMSTI-CG M MACEDONIA
12350 RESEARCH PARKWAY
ORLANDO FL 32826-3726

US GOVERNMENT PRINT OFC
DEPOSITORY RECEIVING SECTION
ATTN MAIL STOP IDAD J TATE
732 NORTH CAPITOL ST NW
WASHINGTON DC 20402

US ARMY RSRCH LAB
ATTN AMSRD-ARL-CI-OK-TP TECHL
LIB T LANDFRIED (2 COPIES)
ABERDEEN PROVING GROUND MD
21005-5066

DIRECTOR
US ARMY RSRCH LAB
ATTN AMSRD-ARL-RO-EN W D BACH
PO BOX 12211
RESEARCH TRIANGLE PARK NC 27709

US ARMY RSRCH LAB
ATTN AMSRD-ARL-CI-CN B NGUYEN
(2 COPIES)
ATTN AMSRD-ARL-CI-CN G RACINE
(2 COPIES)
ATTN AMSRD-ARL-CI-OK-T TECHL
PUB (2 COPIES)
ATTN AMSRD-ARL-CI-OK-TL TECHL
LIB (2 COPIES)
ATTN AMSRD-ARL-D J M MILLER
ATTN AMSRD-ARL-IS-CB P BUDULAS
(2 COPIES)
ATTN IMNE-ALC-IMS MAIL &
RECORDS MGMT
ADELPHI MD 20783-1197

INTENTIONALLY LEFT BLANK.